REMARKS

In response to the final Office Action mailed October 2, 2007, Applicant respectfully requests reconsideration. Claims 1, 3-15 and 17-23 were previously pending in this application. By this amendment, claims 1, 12, 13 and 14 have been amended. No claims have been canceled and no claims have been added. As a result, claims 1, 3-15 and 17-23 are pending for examination with claims 1 and 14 being independent.

I. Rejections Under 35 U.S.C. §103

Claims 1, 3-5, 7, 10, 12-15, 17, 18 and 21 are rejected under 35 U.S.C. §103(a) as purportedly being unpatentable over Mahalingaiah (US Patent No. 5,960,467) in view of Gandhi (U.S. Patent No. 6,405,305) and in further view of Meier (US Patent No. 6,405,305). Applicant respectfully traverses this rejection. While Applicant does not believe the combination of Mahalingaiah, Gandhi and Meier is proper, Applicant believes the claims, as amended, distinguish over the combination and therefore will not argue the appropriateness of the combination herein. However, Applicant reserves the right to argue that the combination is improper at a later time if deemed necessary.

In the "Response to Arguments" section, the final Office Action asserts that Applicant's basis for argument that the claims distinguish over the alleged combination is that the claims require storing a speculative data address in the register file with a one-to-one correspondence to each pipeline stage. What Applicant is arguing, however, is that the claims require that the speculative register file have enough locations to store a speculative address for each of the pipeline stages plus an additional location to store an architectural data address. Although Applicant believes the claims, as previously presented, were clear and distinguish over the alleged combination, independent claims 1 and 14 have been amended to more clearly point out that the *number* of locations for storing speculative data addresses must be equal to or greater than the *number* of stages in the pipeline. The claims do not require that there be any particular correspondence or mapping between the pipeline stages and the speculative register file locations, only that the speculative register file be capable of simultaneously storing at least a *number* of speculative data addresses equal to or greater than the plural number of pipeline stages, plus at least one architectural data address.

The Office Action asserts that the specification does not support storing a speculative data address in the register file with a one-to-one correspondence to each pipeline stage. As discussed above, the claims do not require any particular correspondence, only that the speculative register file be capable of holding the specified number of speculative data addresses (i.e., equal to or greater than the number of pipeline stages), plus at least one architectural data address. The specification clearly supports this limitation. For example, page 7 line 24 – page 8, line 4 of Applicant's specification state:

> Speculative register 230 is configured with sufficient capacity to store the speculative addresses associated with each pipeline stage in execution unit 220 and preferably has additional capacity to permit storage of one or more architectural addresses. *In one embodiment, execution unit 220 has four pipeline stages, and speculative register file 230 has six locations, or slots.* In this embodiment, speculative register file 230 can store four speculative addresses corresponding to the four pipeline stages and up to two architectural addresses (emphasis added).

As a second example, Applicant's specification states on page 9, line 24 – page 10 line 1:

> The in spec register 400 includes one location corresponding to each pipeline stage in execution unit 220 and a commit location. Thus, for the example of a four-stage execution unit, in spec register 400 includes locations 400a, 400b, 400c, 400d and 400e.

Accordingly, the specification includes at least two embodiments providing support for the limitation related to the capacity of the speculative register file. In particular, the above two excerpts disclose a speculative register capable of storing a number of speculative addresses equal to or greater than the number of pipeline stages in the execution unit, plus the capacity to hold at least one additional architectural address. Therefore, Applicant believes the claims are clear and fully supported by the specification.

The Office Action also asserts that page 7, line 24 to page 8 line 8, which is partially reproduced above, indicates a speculative address corresponding to each pipeline *execution stage*, rather than each pipeline stage. However, the cited excerpt clearly refers to the number of pipeline stages, not the number of execution stages. Indeed, there is no mention at all about the

number of execution stages in the pipeline. Applicant respectfully points out that the *execution unit* is not equivalent to an execution stage. The execution unit includes the instruction pipeline and the various pipeline stages that form the pipeline. The instruction pipeline may include one or more execution stages, but it also includes other stages of the instruction pipeline. Accordingly, the claims are believed to be clear and fully supported by the specification in this respect, and to include limitations not disclosed or suggested by the alleged combination, as discussed in further detail below.

Mahalingaiah discloses a conventional speculative register for storing register values for instructions that have been decoded, but have not yet been executed. Mahalingaiah discloses in column 3, lines 49-53 that a "speculative register file stores speculative register values corresponding to previously decoded instructions. The speculative register values are generated prior to execution of the previously decoded instructions." Similarly, in column 7, lines 7-9, Mahalingaiah states that "[s]peculative register file 35 is provided for generating and storing registers values prior to execution of the instructions which generate those values." Mahalingaiah nowhere discloses or suggests that the speculative register file stores anything but register values relating to instructions that have yet to be executed, that is, register values that are speculative.

The only disclosure in Mahalingaiah related to the size of the speculative register file appears in column 18, line 65 – column 19, line 5, which states that "the number of storage locations within register storage 80 is equal in number to the number of lines within reorder buffer 32," which is related to the number of instructions being concurrently decoded (see FIG. 4, 4A and 4B and the corresponding description in columns 18-22). Mahalingaiah describes a speculative register file to store speculative register values, but nowhere mentions anything about the speculative register file storing one more architectural addresses, nor does it disclose a speculative register file having a number of storage locations as recited in independent claims 1 and 14. Gandhi makes no reference to speculative register files and does not cure the deficiencies of Mahalingaiah in this respect.

The Office Action concedes that the combination of Mahlingaiah and Gandhi fails to disclose the result of the speculative registers once they become committed to an architectural state. However, the Office Action asserts that "Meier discloses using pointers and renaming

schemes to allow a speculative register to become architectural." (Page 4 of the Office Action). However, even if Meier does disclose a pointer renaming scheme to label a speculative register as an architectural register, this disclosure does not make obvious the claims as amended herein. In particular, the claims recite a speculative register file capable of storing a number of speculative data addresses equal to or greater than the number of stages in the instruction pipeline, and capable of storing at least one architectural data address in addition.

For example, claim 1 recites "a speculative register file capable of simultaneously storing at least a number of speculative data addresses equal to or greater than the plural number of pipeline stages and at least one architectural data address, the speculative register file configured to hold the speculative data addresses as corresponding instructions advance through the instruction pipeline and at least one speculative data address after the at least one speculative data address becomes a respective architectural data address." That is, the speculative register file can hold a number of speculative data addresses equal to or greater than the number of stages in the pipeline, plus one or more additional architectural data addresses while still being capable of storing the number of speculative data addresses equal to or exceeding the number of pipeline stages.

Meier does not disclose maintaining a speculative data address in number equal to or greater than the number of pipeline stages, even after committing one or more data addresses to architectural data addresses. Rather, Meier discloses converting one of the speculative registers into an architectural state. This technique actually reduces the number of speculative registers available for storing speculative data addresses. Meier is completely silent with respect to providing a speculative register file having a speculative register capable of storing a number of speculative addresses equal to or greater than the number of pipeline stages and capable of simultaneously storing one or more additional architectural addresses. As discussed above (and conceded by the Office Action), Mahalingaiah and Gandhi also fail to disclose this limitation.

A. Claims 1 and 3-13

The alleged combination of Mahalingaiah, Gandhi and Meier does not disclose or suggest a digital signal processor having "a speculative register file capable of simultaneously storing at least a number of speculative data addresses equal to or greater than the plural number of pipeline stages and at least one architectural data address, the speculative register file configured

to hold the speculative data addresses as corresponding instructions advance through the instruction pipeline and at least one speculative data address after the at least one speculative data address becomes a respective architectural data address," as recited in claim 1. Therefore, claim 1 patentably distinguishes over the alleged combination and is in allowable condition.

Claims 3-13 depend from claim 1 and are allowable based at least on their dependency.

B. Claims 14, 15 and 17-23

The alleged combination of Mahalingaiah, Gandhi and Meier nowhere discloses or suggests holding speculative data addresses in a speculative register file as a corresponding instruction advances through the pipeline, "the speculative register file capable of storing a number of speculative data addresses equal to or greater than the plural number of pipeline stages in the pipelined execution unit and at least one architectural data address," as recited in claim 14. Therefore, claim 14 patentably distinguishes over the combination and is in allowable condition. Claims 15 and 17-23 depend from claim 14 and are allowable based at least on their dependency.

In view of the foregoing, Applicant respectfully requests that the rejection under 35 U.S.C. §103 be withdrawn.

CONCLUSION

A Notice of Allowance is respectfully requested. The Examiner is requested to call the undersigned at the telephone number listed below if this communication does not place the case in condition for allowance.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicant hereby requests any necessary extension of time. If there is a fee occasioned by this response, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 23/2825.

Dated: January 2, 2008

Respectfully submitted,

By: William R. McClellan

William R. McClellan
Registration No. 29,409
Wolf, Greenfield & Sacks, P.C.
600 Atlantic Avenue
Boston, Massachusetts 02210-2206
Telephone: (617) 646-8000